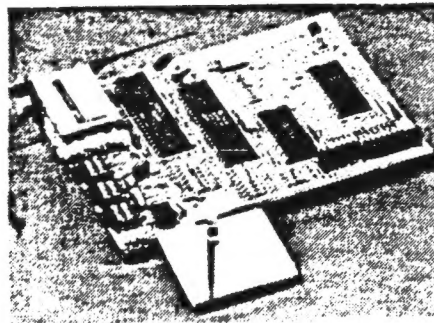
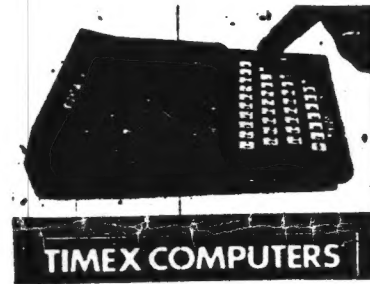


# LISTing Newsletter

Newsletter of the Long Island  
Sinclair/Timex Users Group

Next Meeting

MAY 7, 1995



Game  
Program  
Issue

## *Listing Policy*

*Annual Dues \$16.00*

One "sample" copy sent upon receipt of Business size SASE.  
Copies provided on EXCHANGE BASIS with other bona fide user  
groups. LISTing is published monthly except July and August by  
LIST (Long Island Sinclair Timex) Group, a not for profit user group.

We are always looking for articles, programs, reviews etc. to keep our  
members informed and entertained. You maintain full credit and copyright.

Portions of this publication may be reproduced, without written consent.  
Please give credit to LIST when reprinting articles.

LIST disclaims any responsibility for any damage you may do to your  
computer as a result of reading any articles in LISTing.

\*\*\*\*\*  
L.I.S.T. OFFICERS  
-----

PRES. HARVEY RAIT  
VICE PRES. BOB GILDER  
COR.SECY. JOHN PAZMINO  
ASSOC EDITORS FRED STERN  
HARVEY RAIT  
PUBLISHER BOB GILDER  
LIBRARIAN TOM SKAPINSKY  
TREASURER BOB MALLOY

\*\*\*\*\*  
PLEASE SEND ALL INQUIRIES AND  
SUBMISSIONS(INCLUDING DUES)  
TO; L.I.S.T.

HARVEY RAIT  
5 PERI LANE  
VALLEY STREAM,N.Y.11581

\*\*\*\*\*  
COMING EVENTS:THE NEXT L.I.S.T.  
MEETING WILL BE SUNDAY MAY 7  
AT 2 P.M.AT THE HOME OF HARVEY  
RAIT (SEE ADDRESS ABOVE).

\*\*\*\*\*  
REPORT ON THE MEETING OF APRIL  
9:

THE MEETING STARTED ON TIME AT  
2:00 P.M. WITH 4 MEMBERS PRES-  
ENT.

ONE NEW MEMBERSHIP CAME IN AND  
WE TAKE THIS OPPORTUNITY TO  
WELCOME KENNETH HARBIT OF  
FRESNO CALIFORNIA.

ONE INQUIRY WAS RECIEVED FROM  
ERIC HILL OF MILWAUKEE  
WISCONSIN,WHOSE QUESTIONS WILL  
BE ANSWERED.

WE DECIDED TO KEEP THE MEETING  
INFORMAL AND WOUND UP HAVING A  
ROUNDTABLE DISCUSSION OF  
VARIOUS TOPICS INCLUDING THE  
OBVIOUS MERITS OF THE SINCLAIR  
IN COMPARISON TO THE CURRENT  
PLETHORA OF SO CALLED ADVANCES  
IN THE COMPUTER WORLD TODAY.  
IT TURNS OUT THAT SO MANY OF  
THE CURRENT BALLYHOOD ADVANCES  
IN THE PC WORLD TODAY ARE  
MERELY VARIATIONS OF THINGS  
THAT WE HAVE BEEN DOING FOR  
YEARS WITH OUR TOYS.

BOB GILDER GAVE A DEMONSTRATION  
OF THE "EDITOR" PROGRAM TO SHOW  
HOW TO PRINT OUT A TWO COLUMN  
PAGE IN ONE PASS SIMILAR TO  
THIS PAGE.IT WAS A LITTLE  
COMPLICATED TO A NOVICE THAT  
NEVER USED THIS PROGRAM  
BEFORE,BUT BOB WHO USES IT ALL  
THE TIME MADE IT LOOK EASY.  
MEETING ADJOURNED AT 4:15.

\*\*\*\*\*

FIRST OFF I WANT TO REPEAT THAT THE  
NEXT MEETING WILL BE ON MAY 7,WHICH  
IS THE FIRST SUNDAY OF THE MONTH  
INSTEAD OF THE SECOND SUNDAY BECAUSE  
OF MOTHERS DAY FALLING ON THE 14 th.

-----  
IN GOING THROUGH MY "OLD STUFF" I  
CAME ACROSS A LABORATORY EXPERIMENT  
FROM MY COLLEGE DAYS WHICH WAS MY  
FIRST EXPOSURE TO COMPUTERS.  
THE COOPER UNION DEPT. OF ELECTRICAL  
ENGINEERING,DATED 3/27/1961.TOPIC -  
ELECTRONIC ANALOGUE COMPUTER.  
THE PARAMETERS WERE SET UP IN  
INCREMENTAL STEPS TO PRODUCE RESULTS  
ON A STRIP CHART RECORDER.THE  
PROBLEM CHOSEN WAS TO ANALYZE THE  
EFFECT OF THE DAMPING COEFFICIENT OF  
AN AUTOMOBILE'S SHOCK ABSORBER ON  
THE AUTOMOBILES TRANSLATIONAL MOTION  
WHEN RIDING OVER A RECURRENT BUMP.  
THE PARAMETERS WERE THAT THE BUMP  
WILL BE EQUIVALENT TO A DISPLACEMENT  
OF 0.2 ft AND LAST LAST FOR A  
DURATION OF 0.01 sec.

DIFFERENTIAL EQUATIONS WERE SET UP  
USING SUCH PARAMETERS AS THE MASS OF  
THE AUTO,THE MASS OF THE TIRE AND  
AXLE,THE LINEARIZED SPRING CONSTANT  
OF A PNEUMATIC TIRE,AND THE  
CONSTANTS ASSOCIATED WITH THE SHOCK  
ABSORBER DAMPING CONSTANT-'C1'.  
CALIBRATED AMPLIFIERS WERE SET UP  
AND AN INPUT SIGNAL APPLIED.OUTPUTS  
WERE FED TO A STRIP RECORDER AND  
ANALYZED.

ALL IN ALL THIS WAS A COMPLICATED  
SET-UP TO OBTAIN RESULTS.  
WHY? YOU MAY ASK AM I WRITING ABOUT  
THIS.I GUESS YOU WOULD HAVE HAD TO  
BE AROUND IN THE 50'S AND EARLY 60'S  
TO FULLY APPRECIATE THE BIRTH AND  
MATURATION OF THE DIGITAL COMPUTER  
AS WE NOW USE IT.THE 2K TS-1000  
COULD HAVE SOLVED THIS PROBLEM FAR  
MORE EFFICIENTLY AND EXPEDITIOUSLY  
THAN THE HOURS OF PREPARATION  
REQUIRED TO SOLVE THE STATED PROBLEM  
THAT WERE REQUIRED WITH AN ANALOGUE  
COMPUTER.

A 16 YEAR OLD HACKER CAN NOW DO WHAT  
ONCE REQUIRED YEARS OF STUDIES IN  
PHYSICS AND DIFFERENTIAL ALGEBRA CAN  
WE REALLY LONG FOR THE MORE SIMPLE  
GOOD OLD DAYS?I DONT KNOW THE ANSWER  
BUT I STILL KIND OF LOOK BACK WITH  
NOSTALGIA AND THINK OF CAR RADIOS  
THAT USED A VIBRATOR AND AN OZ4 TO  
GENERATE DC VOLTS.

BY THE WAY I DID GET AN "A" IN THE  
COURSE.

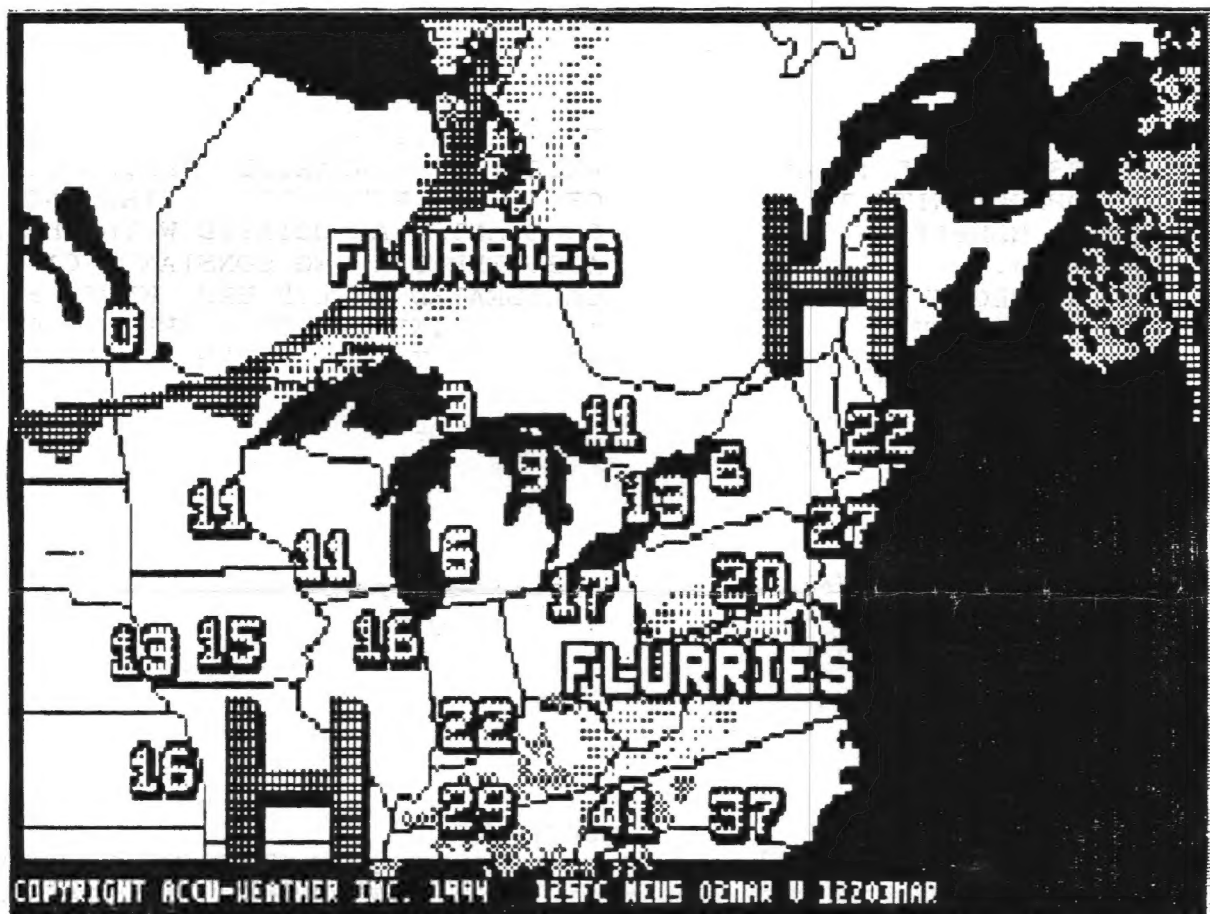
## QL CORNER

During the past two months I have been reviewing some of the IQLR Freeware, Public Domain, Shareware Software which I had ordered from Bob Dyl at \$2.50 each. IQLR-013, UNGIF allows you to review and save onto a QL diskette, GIF format screens which can be located on many bulletin boards which we can download into our QL's. The software disk contains several weather maps in the GIF format that you can load into the UNGIF utility.

The ENGif utility converts partial or complete QL screens to GIF files.

The UNGIF utility requires several Pointer Interface files for this utility which ARE NOT supplied; HOT\_REXT, MENU\_REXT, WMAN and PTR\_GEN. A five page manual accompanies the UNGIF utility and a two page manual for the ENGIF utility. Both manual files have a 'txt' extension. If you don't have a text editor you can import the files into Quill.

To use the Ungif utility 'lrun' the BOOT file which is chained to BOOT\_UNgif. The Ungif\_BOOT stops at line 50, the last line in this file. Line 50: 50 EX\_ungif; drive\_name. Delete the drive\_name and enter FLPL\_ (or any other drive designator) Neast2\_gif (one of the gif files supplied with the program disk). Run the BOOT\_UNgif. Within several seconds the weather map for the North East section of the USA appears (I believe). See the screen dump of this map.



At this time, you should depress either the ESC, SPACE or ENTER key to enter the Main menu. (See the Main Menu screen dump). There are ten (10) Main Menu functions which can be activated with UP/DOWN cursor keys, pressing a letter at the right side of each function or a mouse if you have the mouse interface or serial mouse.

The first function on the menu, QL screen mode : 4-colour. Pressing Space will toggle this line for 8-colour and again monochrome. After setting this function to another mode, say 8-colour, press 'R' (redraw

GIF) and the screen will be redrawn with the 8-colour tones. Experiment with each of the three modes for the best picture display.

The next function, T - Type of picture: monochrome can be changed to color or greytone by pressing the Space bar or a 'Hit'. Again, experiment, using each of these modes by redrawing the picture.

The third function L - Stipple level : 0 = 1x1 can be changed to five different levels including 'dither' which provides 9 greytones, 64 intensity levels or 1000 different stipple-levels in mode 8. Dither provides images of excellent quality and smoothness. It's really a fuzzy method to render a large number of intensities.

V - Save-file type : SCR screen: Toggling the space bar will set the screen extension to SCR, GIF or PIC.

W - Edit the window defination: must be a CON\_ device. CON\_300 becomes 300x180a32x16.

X - Manual move and resize, redraw: Using the cursor keys (or mouse) will allow the screen size to be altered, then redrawn to the new size.

S - Save screen as SCR-file: when this function is activated, delete the screen name as appears and enter a new name and the 'scr' extension.

Z - Use screen as wall-paper: I have not yet used this function!

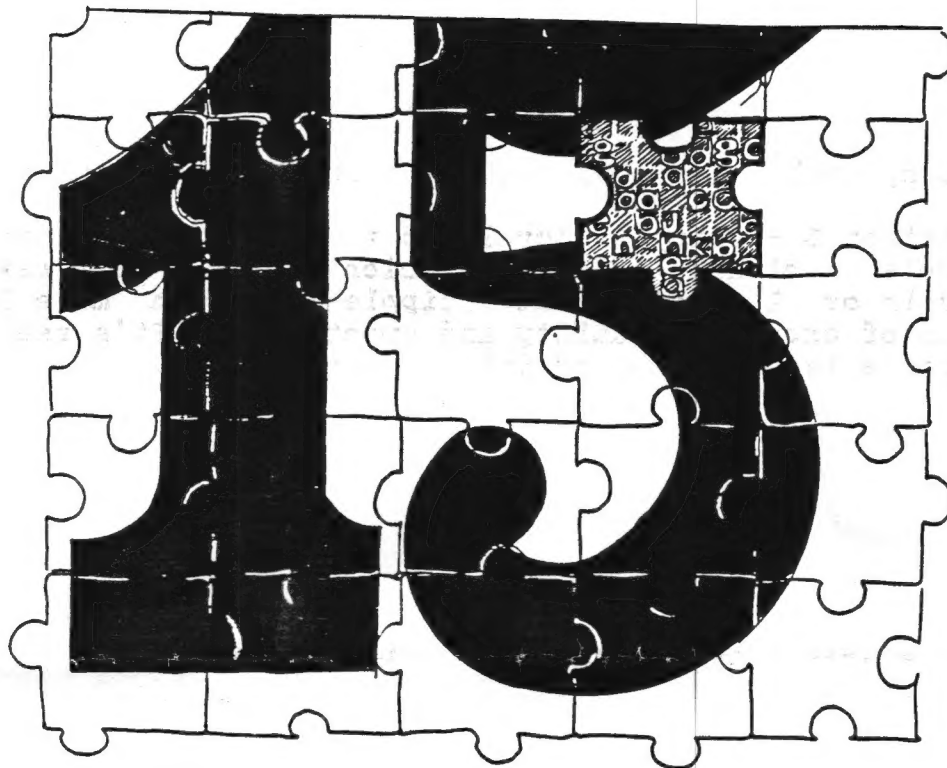
So, there you have it! If you are into developing a large screen shot library, then this program can be a blessing for you. At \$2.50 for the program, it's a steal.



See you next month... Bob Gilder



TS1000  
TS2068



# Puzzle

THE DISPLAY will show a grid of 16 squares. The grid contains the first 15 letters of the alphabet and one space. Slide the letters around in the grid, using the space,

and the puzzle is solved when the letters are in their correct order.

The letters are moved with the usual cursor controls and a count is kept of the moves taken. Your task is

to complete the re-arrangement in as few moves as possible. A diverting little routine, worth having on tape. Submitted by W G Davies of Hereford for the 16K Spectrum.

```

10 REM "15 PUZZLE"
20 REM © W G Davies 1982
100 PRINT AT 8,11;"15 PUZZLE":A
T 20,6;"Choose your colours"
110 LET a$="QEMKLUOGAHFDNIB "
120 LET l=1000: DIM b$(16)
130 LET c$="ABCDEFGHIJKLMNO "
140 INPUT "border ";a,"ink ";b
150 BORDER a: INK b: PAPER 7
170 CLS: FOR c=80 TO 176 STEP
24
180 PLOT c,40: DRAW 0,96: NEXT
c
190 FOR d=40 TO 136 STEP 24: PL
OT 80,d: DRAW 96,0: NEXT d
200 LET j=0: LET k=1: LET p=15:
LET q=20
220 FOR n=6 TO 15 STEP 3: FOR m
=11 TO 20 STEP 3: PRINT AT n,m;2
$(k)
230 LET k=k+1: NEXT m: NEXT n
240 GO SUB 490
250 LET r=p: LET s=q: IF INKEY$
=" " THEN GO TO 250
260 IF INKEY$="5" THEN LET s=q+
3: IF s>20 THEN GO TO 250
270 IF INKEY$="8" THEN LET s=q-
3: IF s<11 THEN GO TO 250
280 IF INKEY$="6" THEN LET r=p-
3: IF r<6 THEN GO TO 250
290 IF INKEY$="7" THEN LET r=p+
3: IF r>15 THEN GO TO 250
300 IF INKEY$<>"5" AND INKEY$<>
"8" AND INKEY$<>"7" AND INKEY$<>
"6" THEN GO TO 250
310 PAPER 7: BEEP .1,0: PRINT A
T p,q;SCREEN$(r,s);AT r,s;" "

```

```

320 GO SUB 400: LET p=r: LET q=
s
330 LET j=j+1: PRINT AT 3,15;j
350 GO TO 250
400 LET a=1: FOR n=6 TO 15 STEP
3: FOR m=11 TO 20 STEP 3: LET b
$(a)=SCREEN$(n,m)
410 LET a=a+1: NEXT m: NEXT n
420 IF b$=c$ THEN GO TO 440
430 RETURN
440 PRINT AT 19,12;"CORRECT": F
OR b=1 TO 16: BEEP .5,b: NEXT b
450 PRINT AT 21,2;"Press any ke
y to play again."
460 IF INKEY$="" THEN GO TO 460
470 PRINT AT 21,2;"
480 PRINT AT 19,12;" "": G
O SUB 600: GO TO 200
490 FOR y=1 TO 50
500 LET r=p: LET s=q: LET x=INT
(RND*4)+5
510 IF x=5 THEN LET s=q+3: IF s
>20 THEN GO TO 500
520 IF x=8 THEN LET s=q-3: IF s
<11 THEN GO TO 500
530 IF x=6 THEN LET r=p-3: IF r
<6 THEN GO TO 500
540 IF x=7 THEN LET r=p+3: IF r
>15 THEN GO TO 500
550 PRINT AT p,q;SCREEN$(r,s);
AT r,s;" "
560 LET p=r: LET q=s: NEXT y: B
EEP .5,0: BEEP .5,5
570 RETURN
600 IF j<1 THEN LET l=j
610 PRINT AT 3,15;"0 "":AT 1,
5;"Best score so far:";l;" "
RETURN

```



# SPACMAN III

For TS2068  
With ROM-SWITCH

**W**E FEEL as if we are lost in a trackless maze, pursued by vengeful ghosts and forced to consume endless quantities of your Spacman programs — but we like it.

The latest breakthrough is from Duncan Anderson of Bishops Frome, Worcester, who has managed to incorporate power pills and intelligent ghosts, which pursue you alternately with carnivorous cunning or flee from you with the agility born of terror. The maze is suitably spaghetti-like, Mr Pacman's jaws face forward, there are exits, and a novel feature is the bomb you can leave behind you if the ghost's icy breath is too close on your collar.

You have only one life in **Spacman III** and there are only two ghosts which do not return to the centre when killed. An additional problem is that the ghost's direction finder becomes confused by corners. A possible way round the problem is to use only straight lines to form the walls of the maze, as do some of the commercial versions of the game.

We would say this listing is a big leap forward and we feel sure that there must be some Clive Sinclair clone who can put all the pieces together and supply us with the real thing (48K Spectrum). Graphics notes:

88,540 etc—Graphic R.  
110—Dots are graphic.  
420, etc—Graphic P.  
510, etc—Graphic L.  
520, etc—Graphic D.  
530, etc—Graphic U.  
2740, etc—Graphic G.

```

5 LET 1000=0
10 RESTORE : DATA 0,0,0,BIN 00
011000,BIN 00011000,0,0,0,0,BIN
1000010,BIN 11100111,255,255,BIN
1111110,BIN 111100,BIN 11000,BIN
N 00001100,BIN 111110,BIN 11111
00,BIN 1111000,BIN 1111000,BIN 1
111100,BIN 111110,BIN 111100
12 DATA 0,BIN 00011000,BIN 001
11100,BIN 01111110,255,255,BIN 1
1100111,BIN 01000010
14 DATA BIN 00111000,BIN 01111
100,BIN 00111110,BIN 00011110,BI
N 00011110,BIN 00111110,BIN 0111
1100,BIN 00111000
16 DATA BIN 00011110,BIN 00111
100,BIN 01111110,BIN 10010011,BI
N 10010011,255,255,BIN 11010101
18 DATA 0,BIN 01111110,BIN 010
0010,BIN 01011010,BIN 01011010,
BIN 01000010,BIN 01111110,0
20 FOR n=0 TO 7: READ x: POKE
USR "0"+n,x: NEXT n
30 FOR n=0 TO 7: READ x: POKE
USR "U"+n,x: NEXT n
40 FOR n=0 TO 7: READ x: POKE
USR "A"+n,x: NEXT n
50 FOR n=0 TO 7: READ x: POKE
USR "S"+n,x: NEXT n
60 FOR n=0 TO 7: READ x: POKE
USR "I"+n,x: NEXT n
70 FOR n=0 TO 7: READ x: POKE
USR "J"+n,x: NEXT n
80 FOR n=0 TO 7: READ x: POKE
USR "P"+n,x: NEXT n
85 GO SUB 9000
88 LET 88="C": LET sc=0: LET b
on=0: LET x=20: LET v=8: LET 9x1
=10: LET 9x2=10: LET 9v1=11: LET
9v2=21
90 DIM b$(21,31)
100 LET b$(1)=
110 LET b$(2)=
120 LET b$(3)=
130 LET b$(4)=
140 LET b$(5)=
150 LET b$(6)=
160 LET b$(7)=
170 LET b$(8)=
180 LET b$(9)=
190 LET b$(10)=
200 LET b$(11)=
210 LET b$(12)=
220 LET b$(13)=
230 LET b$(14)=
240 LET b$(15)=
250 LET b$(16)=
260 LET b$(17)=
270 LET b$(18)=
280 LET b$(19)=
290 LET b$(20)=
300 LET b$(21)=
340 LET 8x=0: LET 8y=000
350 LET 8p=0
355 LET 8x=0

```



# SPACMAN

## III



```
350 LET b=0
370 LET PPX=INT (RND*20+1): LET
PPY=INT (RND*20+1)
380 IF b$ (PPX,PPY)="" THEN GO
TO 370
400 BORDER 1: PAPER 7: INK 1: C
LS
410 PRINT " "
FOR n=1 TO 21: PR
INT " "
NEXT n
420 PRINT AT PPX,PPY: INK 2: "B"
AT 10,1: INK 6: PAPER 1: "B": AT
10,31: INK 6: PAPER 1: "B"
500 IF INKEY$="" THEN GO TO 550
510 IF INKEY$="5" THEN LET b$="
520 IF INKEY$="5" THEN LET b$="
530 IF INKEY$="7" THEN LET b$="
540 IF INKEY$="8" THEN LET b$="
545 IF b=0 AND INKEY$="0" THEN
GO SUB 7000
550 PRINT AT x,y: " ": LET b$(x,
y)="
560 IF b$="V" AND b$(x-1,y)=""
THEN LET x=x-1
570 IF b$="A" AND b$(x+1,y)=""
THEN LET x=x+1
580 IF b$="E" AND b$(x,y+1)=""
THEN LET y=y+1
585 IF b$="3" AND b$(x,y-1)=""
THEN LET y=y-1
590 PRINT INK 0: AT x,y: b$
600 IF b$(x,y)="" THEN LET sc=
sc+1: BEEP .015,5: PRINT AT 0,2:
sc+b0n
610 IF sc=342 OR sc=784 OR sc=1
220 OR sc=1060 THEN GO TO 4000
620 IF 9x1=9x2 AND 9y1=9y2 THEN
635 PRINT AT 9x2,9y2: b$(9x2,9y2
)
640 PRINT AT 9x1,9y1: b$(9x1,9y1
)
641 IF RND<.2 THEN FOR n=0 TO 1
0: NEXT n: GO TO 700
650 IF b$(9x1+1,9y1)="" AND 9
x1<x THEN LET 9x1=9x1+1
655 IF b$(9x2-1,9y2)="" AND 9
x2>x THEN LET 9x2=9x2-1
660 IF b$(9x1-1,9y1)="" AND 9
x1>x THEN LET 9x1=9x1-1
665 IF b$(9x2+1,9y2)="" AND 9
x2<x THEN LET 9x2=9x2+1
670 IF b$(9x1,9y1+1)="" AND 9
y1<y THEN LET 9y1=9y1+1
675 IF b$(9x2,9y2+1)="" AND 9
y2<y THEN LET 9y2=9y2+1
680 IF b$(9x1,9y1-1)="" AND 9
y1>y THEN LET 9y1=9y1-1
685 IF b$(9x2,9y2-1)="" AND 9
y2>y THEN LET 9y2=9y2-1
740 PRINT AT 9x1,9y2: INK 3: "B"
745 PRINT AT 9x2,9y1: INK 3: "B"
746 IF PP=0 THEN PRINT AT PPX,PP
Y: INK 2: "B"
748 IF b=1 THEN PRINT AT bx,by:
INK 4: "B"
750 IF (9x1=x AND 9y1=y) OR (9x
2=x AND 9y2=y) THEN GO TO 1000
760 PRINT AT 0,2: sc+b0n
765 IF 9x>5 THEN GO TO 780
770 IF x=10 AND y=2 AND b$="3"
THEN PRINT AT x,y: " "
BEEP .1,24: LET v=30: BORDER 1:
GO TO 775
775 IF x=10 AND y=30 AND b$="6"
THEN PRINT AT x,y: " "
BEEP .1,24: LET v=2: BORDER 1:
GO TO 770
779 LET ex=ex+1: IF ex>5 THEN P
```

```
PRINT AT 10,1: " ": AT 10,31: " "
780 IF x=PPX AND y=PPY AND PP=0
THEN LET PP=1: GO SUB 2000
790 IF b=1 AND ((9x1=bx AND 9y1
=by) OR (9x2=bx AND 9y2=by)) THE
N GO SUB 3000
1000 PRINT AT x,y: INK 0: PAPER
6: FLASH 1,5: FOR n=24 TO -24: 3
TEP -1: BEEP .05,n: NEXT n
1002 BORDER 0: PAPER 0: INK 0: C
LS: PRINT AT 10,11: FLASH 1: "SC
ORE: "
1004 FOR n=0 TO 100: BEEP .01,n:
50: NEXT n
1005 LET 1090=1: GO TO 0
1006 LET b0n=b0n+5: BORDER 2: FO
R n=10 TO 20: BEEP .05,n: NEXT n
2100 FOR v=0 TO 20
2500 IF INKEY$="" THEN GO TO 255
0
2510 IF INKEY$="5" THEN LET b$="
2520 IF INKEY$="5" THEN LET b$="
2530 IF INKEY$="7" THEN LET b$="
2540 IF INKEY$="8" THEN LET b$="
2550 PRINT AT x,y: " ": LET b$(x,
y)="
2560 IF b$="V" AND b$(x-1,y)=""
THEN LET x=x-1
2570 IF b$="A" AND b$(x+1,y)=""
THEN LET x=x+1
2580 IF b$="E" AND b$(x,y+1)=""
THEN LET y=y+1
2585 IF b$="3" AND b$(x,y-1)=""
THEN LET y=y-1
2590 PRINT AT x,y: INK 2: b$
2600 IF b$(x,y)="" THEN LET sc=
sc+1: BEEP .01,20: PRINT AT 0,2:
sc+b0n
2635 PRINT AT 9x2,9y2: b$(9x2,9y2
)
2640 PRINT AT 9x1,9y1: b$(9x1,9y1
)
2641 IF RND<.3 THEN FOR n=0 TO 1
0: NEXT n: GO TO 2700
2650 IF b$(9x1+1,9y1)="" AND 9
x1<x THEN LET 9x1=9x1+1
2655 IF b$(9x2-1,9y2)="" AND 9
x2>x THEN LET 9x2=9x2-1
2660 IF b$(9x1-1,9y1)="" AND 9
x1>x THEN LET 9x1=9x1-1
2665 IF b$(9x2+1,9y2)="" AND 9
x2<x THEN LET 9x2=9x2+1
2670 IF b$(9x1,9y1+1)="" AND 9
y1<y THEN LET 9y1=9y1+1
2675 IF b$(9x2,9y2+1)="" AND 9
y2<y THEN LET 9y2=9y2+1
2680 IF b$(9x1,9y1-1)="" AND 9
y1>y THEN LET 9y1=9y1-1
2685 IF b$(9x2,9y2-1)="" AND 9
y2>y THEN LET 9y2=9y2-1
2740 PRINT AT 9x1,9y2: INK 3: "B"
2745 PRINT AT 9x2,9y1: INK 3: "B"
2750 IF (9x1=x AND 9y1=y) OR (9x
2=x AND 9y2=y) THEN GO TO 3000
2770 IF x=10 AND y=2 AND b$="3"
THEN PRINT AT x,y: " ": BEEP .01,
24: LET v=30
2775 IF x=10 AND y=30 AND b$="6"
THEN PRINT AT x,y: " ": BEEP .05
,24: LET v=2
2800 NEXT v
2900 BORDER 0: FOR n=0 TO 3: BEE
P .1,0: BEEP .1,1: NEXT n: BORDE
R 1
2910 RETURN
3000 PRINT AT x,y: BRIGHT 1: FLA
SH 1: INK 2: PAPER 0: "B"
3010 FOR b=1 TO 3: FOR n=12 TO 2
```

```
4: BEEP .05,n: NEXT n: NEXT b
3020 LET b0n=b0n+25: PRINT AT 0,
2: sc+b0n
3021 PRINT AT 9x1,9y1: b$(9x1,9y1
): AT 9x2,9y2: b$(9x2,9y2)
3022 LET 9x1=INT (RND*20+1): LET
9y1=INT (RND*20+1): IF b$(9x1,9
y1)="" THEN GO TO 3020
3025 LET 9x2=INT (RND*20+1): LET
9y2=INT (RND*20+1): IF b$(9x2,9
y2)="" THEN GO TO 3020
3028 PRINT AT 9x1,9y1: INK 3: "B"
AT 9x2,9y2: INK 3: "B"
3030 RETURN
4000 LET sc=sc+100: BORDER 0: PA
PER 0: INK 3: CLS
4010 PRINT AT 10,30: INK 2: "B"
4020 FOR x=1 TO 27: PRINT AT 10,
x: INK 3: "3"
4030 BEEP .05: NEXT x
4030 FOR n=12 TO 24: BEEP .01,n:
NEXT n
4040 FOR j=27 TO 1 STEP -1: PRIN
T AT 10,j: INK 4: "B"
4050 BEEP .05,50-j: NEXT
j
4060 GO TO 100
7000 BORDER 4: LET bx=x: LET by=
y: FOR n=0 TO 20: BEEP .01,n: NE
XT n: PRINT AT bx,by: INK 4: "B"
LET b=1: BORDER 1
7100 RETURN
9000 LET b=2: PRINT AT bx,by: FL
ASH 1: INK 4: PAPER 0: "B": FOR n
=0 TO 40: BEEP .01,n: NEXT n: PR
INT AT bx,by: b$(bx,by): LET b0n=
b0n+15
9100 PRINT AT 9x1,9y1: b$(9x1,9y1
): AT 9x2,9y2: b$(9x2,9y2)
9100 LET 9x1=INT (RND*20+1): LET
9y1=INT (RND*20+1): IF b$(9x1,9
y1)="" THEN GO TO 9100
9100 LET 9x2=INT (RND*20+1): LET
9y2=INT (RND*20+1): IF b$(9x2,9
y2)="" THEN GO TO 9100
9200 RETURN
9300 LET 1090=x: LET 9y1=30-y
9305 IF v=1 THEN GO SUB 5000
9310 IF b$(9x1,9y1)="" THEN LET
9y1=9y1-1: GO TO 9305
9300 RETURN
9400 LET x=x+1
9410 IF x=27 THEN LET x=1
9420 IF b$(9x1,9y1)="" THEN RE
TURN
9440 RETURN
9450 IF b$="" SPACMAN 3 BY DUNC
AN ANDERSON
9501 BORDER 0: PAPER 0: INK 0: C
LS
9502 PRINT AT 10,0: "PRESS ANY KE
Y"
9505 PAUSE 0.1 THEN RETURN
9510 FOR n=1 TO 31: BEEP .01,n:
LET b$=b$(1 TO n): PRINT AT 10,3
1-LEN b$: b$: NEXT n
9520 FOR n=0 TO 100: NEXT n
9530 FOR n=0 TO 30: PRINT AT 10,
n: " "
AT 10,n+1: INK 3: BRIGHT 1
n: BEEP .1,10+n: NEXT n
9530 PRINT AT 10,31: INSTRUCTIONS:
AT 5,0: INK 7: "1. AT 7,5: INK 3
"
AT 9,0: INK 2: "2. AT 11,6:
INK 6: "
AT 13,0: INK 4: "3.
FOR n=0 TO 4: PRINT AT 5+2n,
n: "
9560 PRINT AT 5,14: "YOU"
AT 7,14: "GHOST"
AT 9,14: "POWER PILL"
AT 11,14: "EXIT"
AT 13,14: "BOMB (PR
ESS 0)"
9570 FOR n=10 TO 40: BEEP .1,n:
NEXT n
9580 RETURN
```

TS 1000



THE INVISIBLE TOOLKIT is a suite of fast efficient machine-code routines driven by a short Basic menu. Due to the fact that extensive use is made of routines already present in the Rom the machine-code is written in less than 800 bytes, which together with the Basic lines occupies less than 1800 bytes of memory.

The program was written to make it easier to modify and develop a Basic program and even combine two or more Basic programs — or routines — so that the idea of having a library of useful subroutines on cassette is quite feasible. In addition, the facility to create large Rems, or delete them at a stroke, is invaluable when assembling machine-code.

The following routines are available:

**DOWN CODE** — used after Up Code. Downloads the copy in upper memory into a specified line and Address. Space is automatically created.

**DOWN BASIC** — used after Up Basic. Brings the Basic copy down into the Basic area placing it according to the first line number of the block.

**LINE ADDRESS** — gives the address of the specified line number. The first character is four bytes further on.

**REM EXPAND** — expands the Rem statement with the specified line number, from a specified address, by a given number of bytes.

**UP CODE** — copies a given number of bytes from a given address, into upper memory.

## THE INVISIBLE TOOLKIT

Stuart Clarke presents a number of machine-code routines for the ZX-81 which are driven by a short Basic menu.



**UP BASIC** — copies a block of Basic lines specified by first and last lines.

**BLOCK DELETE** — specify first and last lines.

**RENUMBER BASIC** — renumbers a block of Basic lines specified by first and last lines. You are allowed to specify both the Increment and the New first line number. All non-computed Gotos, Gosubs, and Runs will be correctly changed, provided the line number is in four-digit form, e.g., Goto 1 as Goto 0001.

**RENUMBER COPY** — used after Up Basic. Only requires the Increment and New first line number to be specified.

The order in which the routines are listed may seem a little odd, but this is as they appear in the menu. This is dictated by the use of relative rather than absolute jumps in the machine-code to allow it to be position independent, and the entire program has been written in such a way that, as well as the machine code, the Basic menu can be transferred to upper memory — without overwriting existing upper memory contents — to allow the Loading of subsequent programs.

When required, the menu can be recalled, and it will not clash with the existing Basic as its lines are numbered A222 to A239 — 10222 to 10239. This is possible since the Rom checks the validity of a line number by only comparing the high byte with 40 decimal — 40 times 256 equals 10240, the first "non-valid" line number.

However, line numbers in the range 10000 to 11239 are only obtained by using the Renumber routine, as 9999 is the largest line you can enter direct from the keyboard. The menu lines are now tagged on to the end of the program you have just Loaded.

#### Report codes.

K/A234	you have attempted to use Down Code when no copy exists in upper memory.
T/A233	you have attempted to use Down Basic or Renumber Basic when the "live" copy is not Basic.

#### System variables.

16388-16389	Ramtop
16507-16508	Old Ramtop
16444-16451	Poked by menu
ZX81	variable pointing to the first location in the "jump table" accessing the machine-code routines.

If the typing seems a bit of a chore, just send your name and address with a cheque for £3 made payable to Sarglen Software, to 147 South Parks Road, Glenrothes, Fife, KY6 1NT, and I will send you a quality cassette with two copies each for machines with and without the "improved Rom", together with an instruction leaflet.

Type in program 1, then duplicate line 1 using the Edit key so that you have 12 identical lines, numbered 1 to 12. Now Run program 1.

Type in program 2, and Run it. This is the hex loader. Enter the address 16544 — or the

#### Program 1.

```
1 REM .....
20 POKE 16511,68
```

```
38 POKE 16512,3
48 FOR N=16514 TO 17347
58 POKE N,1
68 NEXT N
```

#### Program 2.

```
10 INPUT S
20 LET A$=""
30 GOSUB 0150
40 IF A$="" THEN INPUT A$
50 IF A$="S" THEN STOP
60 LET X=16*CODE A$+CODE A$(2)
-475
70 POKE S,X
80 LET Y=Y+X
90 LET C=C+1
100 LET S=S+1
110 PRINT A$( TO 2); " ";
120 LET A$=A$(3 TO )
130 IF C=5 THEN GOSUB 0150
140 GOTO 0040
150 PRINT TAB 25;Y
160 LET Y=0
170 LET C=0
180 PRINT S;TAB 8;
190 RETURN
```

#### Program 3.

```
100 LET F=16508
20 LET L=764
30 POKE 16444,F-256+INT (F/256)
40 POKE 16445,INT (F/256)
50 POKE 16446,L-256+INT (L/256)
60 POKE 16447,INT (L/256)
70 RAND USR 16767
80 RAND USR 16522
90 LET ZX81=PEEK 16388+256+PEE
16388+253
100 FAST
110 FOR N=0 TO 21
120 PRINT "
130 NEXT N
140 PRINT AT 1,1: "
150 PRINT AT 5,6: "ZX81=";ZX81;A
T 5,6: "PEEK 16507=";PEEK 16507;A
T 7,6: "PEEK 16508=";PEEK 16508
160 PRINT AT 10,1: "
170 PRINT AT 12,6: " RAND USR (Z
X81+2)
180 PRINT AT 15,1: "
190 PRINT AT 17,6: "GOTO 10222"
200 FAST
210 PRINT AT 0,18: "
220 AT 2,0: "1.DN CODE--LINE:AD
```

```
DR: "AT 4,0: "2.DNBASIC--"AT 5,0
"3.LN ADDR--LINE--"AT 8,0: "4.AE
M EXP--LINE--BYTE--ADDR--"AT 10,0
"5.UP CODE--ADDR--BYTE--"AT 12,0
"6.UBASIC--1ST--LAST--"AT 14,0
"7.BLK DEL--1ST--LAST--INCR--NEUL
"8.RENUM--1ST--LAST--INCR--NEUL
"AT 16,0: "9.RENUM.C
INCR: NEUL "AT 21,0: "ENTER NO."
310 INPUT N
320 PRINT AT 2+N,1: "
330 FOR M=1 TO 4
340 INPUT Z
350 PRINT AT 2+N,5+M+5: "
AT 2+N,6+M+4: Z
360 POKE 16442+M+2,Z-256+INT (Z
/256)
370 POKE 16443+M+2,INT (Z/256)
380 NEXT M
390 CLS
400 IF N=2 OR N=9 THEN RAND USR
(ZX81+500)
410 IF N=1 OR N=2 OR N=9 THEN R
AND USR (ZX81+535)
420 LET ZX82=USR (ZX81+2+N-2)
430 IF N=5 OR N=6 OR N=9 THEN R
AND USR (ZX81+501)
440 IF N=3 THEN PRINT "ADDRESS3
OF LINE "PEEK 16444+256+PEEK 16
445: " IS ";ZX82
450 REM
460 REM
```

#### Program 4.

```
2 SAVE "TOOLKIT"
100 FAST
200 POKE 16444,10222-256+INT (1
0222/256)
210 POKE 16445,INT (10222/256)
220 POKE 16446,10239-256+INT (1
```

```
0239/256)
230 POKE 16447,INT (10239/256)
240 RAND USR (ZX81+10)
250 POKE 16444,1
260 POKE 16445,0
270 LET N=7
280 GOTO 10235
```

appropriate address if you are resuming after a break — and enter the hex codes listed in figure 1. You may enter any number of bytes at a time, and the screen display should tally with figure 2. If you make a mistake input "S", which will stop the program. To recommence, simply Goto 10, and enter your new starting address.

When this is complete — and you should have Saved your program at several stages — Save the program thus far.

If your ZX-81 has the "unimproved" Rom, identified by Print Peek 5404 giving 253, then you must enter the five Pokes in fig.2 before Saving and proceeding further.

Type in program 3. This is most of the Basic required to transfer the code to upper memory and provide the operating menu. When you have Saved this, Run it. You should now have the menu on screen, so select 8 — Renumber — and enter the numbers 290, 460, 1, and 10222 in response to the prompts.

Almost immediately, the menu lines will be renumbered from A222 to A239 — 10222 to 10239.

If all is well, type in program 4, the remaining Basic, and Save the program again. You are now ready to Save this, the final completed program in auto-run mode. To do this simply Run the program whilst recording in the usual way. When the program has Saved, you will find that the Basic program area has been cleared and the menu lines are now stored above Ramtop where they will remain until required. Similarly, this occurs when you Load the auto-run version.

The Invisible Toolkit is now ready to use, so refer to the instructions and try it out.

Using the toolkit is easy. After Loading,

RAND USR (ZX81 + 2)

retrieves the Basic from above Ramtop, and Goto 10222 displays the menu. The first

(continued on next page)

(continued from previous page)

prompt requires you to input your selection — 1 to 9 — and the next four prompts expect numerical input, for example line number, Address — or any number if the menu indicates a blank.

When you wish to Load another program

on which to use the toolkit, there are a couple of things to remember. First make sure that the menu lines are back above Ramtop — if not, use Copy Basic to do so. Then note down the contents of locations 16507 and 16508, as you will need to restore them after Loading.

Also, the variable ZX-81 will have to be

restored after Loading — also after Run and Clear, which are best avoided.

After you have LOADED the program you intend to work on, and restored the system variables 16507 and 16508, and ZX-81

RAND USR (ZX81 + 2)

and away you go!

Figure 1.

Figure 1.

16514	00	00	D1	00	53	=292
16512	00	75	75	ED	5B	=564
16524	04	40	ED	53	7B	=511
16529	40	21	01	40	CD	=495
16534	A5	40	D5	3E	02	=506
16539	CD	A5	40	CD	AD	=812
16544	40	3D	20	F7	C9	=605
16549	23	4E	23	46	EB	=453
16554	09	EB	C9	EB	FD	=933
16559	E1	C1	C5	71	23	=763
16564	70	EB	FD	E9	2A	=875
16569	3C	40	CD	D0	09	=554
16574	E5	2A	3E	40	23	=432
16579	CD	D0	09	D1	CD	=844
16584	17	0A	C9	2A	04	=200
16589	40	E5	2A	3E	40	=461
16594	10	13	10	F5	21	=345
16599	10	27	22	2B	40	=196
16604	2A	04	40	E5	CD	=544
16609	0B	05	EB	CD	D0	=840
16614	09	E5	2A	7B	40	=467
16619	ED	5B	04	40	CD	=601
16624	17	0A	E1	E5	C5	=604
16629	2B	CD	9E	09	C1	=606
16634	D1	E1	C5	ED	B0	=1044
16639	C1	C5	2A	02	40	=400
16644	09	22	02	40	ED	=346
16649	73	36	40	2A	36	=329
16654	40	09	F9	A7	ED	=726
16659	42	ED	5B	04	40	=462
16664	EB	A7	ED	52	44	=709
16669	4D	ED	5B	7B	40	=592
16674	2A	04	40	ED	53	=430
16679	04	40	1B	2B	ED	=375
16684	00	10	34	2A	3C	=362
16689	40	CD	D0	09	44	=562
16694	4D	C9	ED	4B	3E	=632
16699	40	2A	40	40	2B	=277
16704	E5	C5	CD	9E	09	=790
16709	C1	E1	C5	41	3E	=742
16714	10	23	77	10	FC	=449
16719	10	12	10	01	10	=219
16724	01	10	D0	10	DF	=616
16729	10	24	10	2B	10	=151
16734	77	10	77	10	77	=405
16739	2A	3C	40	CD	D0	=507
16744	09	C1	20	0B	23	=200
16749	23	5E	23	56	EB	=405
16754	09	EB	72	2B	73	=516
16759	2A	29	40	09	22	=190
16764	29	40	C9	2A	3C	=400
16769	40	ED	40	3E	40	=502
16774	10	03	CD	0B	40	=400
16779	C5	D5	E5	CD	C5	=1041
16784	0E	2A	04	40	22	=150
16789	7B	40	A7	ED	42	=657

16794	22	04	40	C5	E5	=520
16799	2A	02	40	A7	ED	=512
16804	42	22	02	40	ED	=403
16809	73	36	40	2A	36	=329
16814	40	A7	ED	42	F9	=703
16819	09	ED	50	7B	40	=524
16824	EB	A7	ED	52	44	=709
16829	4D	2A	36	40	ED	=474
16834	73	5D	40	ED	5B	=600
16839	5D	40	ED	B0	E1	=795
16844	C1	D1	D5	EB	ED	=1007
16849	00	E1	D1	C1	C9	=1004
16854	10	04	10	1E	10	=106
16859	12	CD	00	40	C5	=660
16864	CD	00	0A	2A	29	=394
16869	40	A7	C1	ED	42	=727
16874	22	29	40	C9	FD	=593
16879	CB	01	EE	FD	CB	=090
16884	01	A6	10	52	FD	=526
16889	CB	01	AE	FD	CB	=034
16894	01	A6	21	7D	40	=309
16899	FD	CB	01	9E	E5	=044
16904	23	23	23	23	7E	=266
16909	FE	FA	20	04	FD	=793
16914	CB	01	DE	7E	FE	=006
16919	76	C1	23	20	1F	=417
16924	20	C5	7E	CD	04	=751
16929	07	7E	FE	EC	20	=663
16934	25	FE	ED	20	21	=601
16939	FE	F7	20	1D	FD	=023
16944	CB	01	5E	23	20	=365
16949	DF	E1	CD	F2	09	=904
16954	EB	ED	5B	0C	40	=639
16959	A7	E5	ED	52	E1	=940
16964	20	04	10	0B	10	=279
16969	05	10	3C	23	23	=336
16974	23	23	23	E5	7E	=460
16979	CD	04	07	20	2D	=469
16984	23	7E	FE	76	20	=565
16989	27	E1	E5	FD	CB	=949
16994	01	06	20	20	23	=011
16999	7E	23	5E	23	56	=376
17004	23	4E	23	46	CD	=423
17009	0D	15	C5	E1	CD	=709
17014	D0	09	10	02	10	=275
17019	0A	E5	C1	E1	2B	=076
17024	70	2B	71	10	B1	=469
17029	D1	10	AE	FD	CB	=063
17034	01	06	20	50	C9	=432
17039	20	56	20	5E	EB	=601

Figure 2.

POKE	17051,20
POKE	17054,244
POKE	17082,20
POKE	17091,20
POKE	17110,201

17044	CD	08	05	42	4B	=530
17049	C5	CD	20	15	CD	=660
17054	F0	13	C1	E1	E5	=914
17059	C5	23	77	23	73	=501
17064	23	72	23	71	23	=332
17069	70	C1	E1	2B	10	=597
17074	02	10	93	10	04	=199
17079	E5	D5	CD	20	15	=700
17084	EF	C0	34	01	0A	=494
17089	00	CD	20	15	EF	=497
17094	C1	05	24	C2	E1	=653
17099	04	E0	01	03	E2	=450
17104	34	CD	A7	0E	C5	=635
17109	CD	CD	15	C1	D1	=833
17114	E1	C6	1C	77	2B	=613
17119	15	20	D5	10	96	=440
17124	10	CD	2A	7B	40	=450
17129	22	46	40	2A	04	=214
17134	40	FD	CB	01	0E	=631
17139	20	10	00	00	00	=56
17144	00	00	00	2A	3E	=104
17149	40	23	CD	D0	09	=529
17154	22	46	40	2A	3C	=270
17159	40	CD	D0	09	00	=494
17164	00	ED	40	40	40	=440
17169	ED	5B	42	40	C5	=655
17174	D5	D1	C1	7A	FE	=991
17179	20	30	03	11	0F	=131
17184	27	72	23	73	2B	=346
17189	EB	09	EB	C5	D5	=009
17194	CD	F2	09	EB	ED	=920
17199	5B	46	40	A7	E5	=621
17204	ED	52	E1	20	DE	=790
17209	D1	D1	FD	CB	01	=075
17214	E6	FD	CB	01	0E	=797
17219	C0	10	9E	21	FE	=661
17224	27	CD	D0	09	23	=504
17229	23	5E	23	56	13	=269
17234	13	72	2B	73	19	=316
17239	20	01	02	00	C5	=243
17244	E5	CD	9E	09	D1	=010
17249	C1	21	7B	40	ED	=650
17254	00	C9	CD	23	0F	=632
17259	21	FE	27	CD	D0	=747
17264	09	23	23	5E	23	=200
17269	56	D5	10	10	72	=467
17274	2B	73	D1	D5	19	=605
17279	7E	FE	EA	20	07	=653
17284	D1	20	72	2B	73	=524
17289	CF	13	D1	01	02	=430
17294	00	C5	11	7C	40	=402
17299	ED	00	23	C1	E5	=070
17304	09	EB	E1	CD	00	=770
17309	0A	C9	2A	04	40	=321
17314	7E	FE	20	30	09	=477
17319	2A	7B	40	2B	7E	=390
17324	FE	76	20	02	CF	=621
17329	1C	C9				

16789

Figure 2.

POKE 17051,20  
POKE 17054,244  
POKE 17082,20  
POKE 17091,20  
POKE 17110,201